CS 202 - Fall 2017  
Dr. Williams - sections 1003, 1004, and 1007  
Assignment 4  
Electronic copy due Monday, November 6 at 11:59 AM  
Hard copy due Monday, November 6 or Tuesday, November 7 at the beginning of your class

In this assignment you will write a class called intdouble that contains an int array of size 10 and a double array of size 10. The initial values of these arrays should be 1, 2, 3, 4... and 1.1, 2.2, 3.3,... respectively.

You are to implement a constructor and the following operator overloads in your class:

- `=` to copy the arrays from one object to another
- `<<` to display the contents of each array, with each value displayed using `setw(10)` and a newline after each array
- `>>` for an integer input, n, set the values of the int array to 1*n, 2*n, 3*n, ... and the values of the double array to 1.1*n, 2.2*n, 3.3*n, ... (see example code below)
- `[]` such that `[n]` returns an alias to the nth element of the int array only. Use `assert` to abort the program if there is attempted access of array elements outside of the appropriate range. Do not forget to implement a `const` version as well.
- `++` and `--` (both pre and post) to increment/decrement each element in the int array only, as appropriate
- 4 arithmetic operators (+ - / and *) such that they perform as described below:
  - Assuming a and b are intdouble objects with default values:
    - `a + b` should result in an intdouble object that contains 2, 4, 6... and 2.2, 4.4, 6.6,...
    - `a - b` should result in an intdouble object that contains 0, 0, 0,... and 0.0, 0.0, 0.0,...
    - `a * b` should result in an intdouble object that contains 1, 4, 9,... and 1.21, 4.84, 10.89,...
    - `a / b` should result in an intdouble object that contains 1, 1, 1,... and 1.0, 1.0, 1.0,...
- All 6 relational operators (`>`, `>=`, `<`, `<=`, `==`, `!=`)  
  - `a > b` iff the sum of all 20 array values in a is greater than those of b
  - `a == b` iff all 20 array values in a are equal to those of b
- Note, this is a total of 19 operator overloading functions.

You must submit a tarfile containing exactly 3 files as follows:

- `loginnname.intdouble.h` - the header, with include guards of
- `loginnname.intdouble.cpp` - the implementation file
- `makefile` - to compile the class into object code and compile `main.cpp` (which I will use to test your code) into an executable. See example on next page.

Notes:
- Change `loginnname` to your exact login name on bobby
- Use `tar -tf yourfile.tar` to verify that exactly the 3 above files are included, then submit with assignment code 14.
- You do not need to provide a main body (I have my own that will test your code). You will, of course, want to write one for yourself so that you can verify that your class works.
In addition to the usual (e.g. not turned in on time, no paper copy, doesn’t compile, not doing information hiding) the following may result in an immediate grade of 0:

- If any of the functions are not implemented.
- If you do not provide a valid makefile that works.
- If you do not name your files appropriately.
- If you do not call your class `intdouble`
- If your tar file does not contain exactly the 3 files and no others.
- If you do not submit using the correct assignment code (14).

---

Sample makefile (must be called `makefile` or `Makefile`):

```make
all:
    g++ -Werror -Wpedantic -Wall -Wextra -std=c++11 -c loginname.intdouble.c
    g++ -Werror -Wpedantic -Wall -Wextra -std=c++11 -o loginname.out main.cpp loginname.intdouble.c
```

You will **not** include `main.cpp` in your tar file, but should write one for your own testing. I have my own `main.cpp` for testing your class.

My example `main.cpp` will be similar in nature to this (although more complicated):

```cpp
#include <iostream>
#include "loginname.intdouble.h"

using namespace std;

int main()
{
    intdouble a, b;

    cout << a;

    if (a < a + b)
        cout << "Whee!" << endl;

    cout >> "Enter a value: ";
    cin >> a;

    cout << a;

    return 0;
}
```

The correct output for this should be:

```
1 2 3 4 5 6 7 8 9 10
1.1 2.2 3.3 4.4 5.5 6.6 7.7 8.8 9.9 10.10
```

Whee!
Enter a value: 3

```
3   6   9  12  15  18  21  24  27  30
3.3  6.6  9.9 13.2 16.5 19.8 23.1 26.4 29.7 33
```